```
#include <pthread.h>
#include <stdio.h>

void* myFunction(void* arg) {
    printf("Thread is running.\n");
    return NULL;
}

int main() {
    pthread_t thread;

    // Create a thread
    if (pthread_create(&thread, NULL, myFunction, NULL) != 0) {
        perror("Thread creation failed");
        return 1;
    }

    // Wait for thread to complete
    pthread_join(thread, NULL);

    printf("Main function done.\n");
    return 0;
}
```

```
#include <pthread.h>
#include <stdio.h>

void* threadFunc(void* arg) {
    printf("Thread says hello!\n");
    return NULL;
}

int main() {
    pthread_t tid;

    pthread_create(&tid, NULL, threadFunc, NULL);

    // Wait for the thread to finish
    pthread_join(tid, NULL);

    printf("Main: thread joined successfully.\n");
    return 0;
}

#include <pthread.h>
#include <stdio.h>

void* func(void* arg) {
    pthread_t tid = pthread_self();

    // Compare with main thread ID
    if (pthread_equal(tid, *(pthread_t*)arg)) {
        printf("Thread IDs are equal.\n");
    } else {
        printf("Thread IDs are different.\n");
    }
    return NULL;
}

int main() {
    pthread_t main_tid = pthread_self();
    pthread_t new_tid;

    pthread_create(&new_tid, NULL, func, &main_tid);
    pthread_join(new_tid, NULL);

    return 0;
}
```

```
#include <pthread.h>
#include <stdio.h>

void* exitFunc(void* arg) {
    printf("Thread exiting early...\n");
    pthread_exit(NULL);  // Exit the thread
}

int main() {
    pthread_t tid;

    pthread_create(&tid, NULL, exitFunc, NULL);
    pthread_join(tid, NULL);
    printf("Main thread waited for exit thread.\n");
    return 0;
}
```